ENGENHARIA REVERSA APLICADA A BANCOS DE DADOS RELACIONAIS

Hiroo Takaoka[*]

RESUMO

Há várias motivações para fazer a engenharia reversa de banco dados (BD): (1) recuperar a descrição do conteúdo do BD que se perdeu ao longo do tempo, devido a modificações para implementar mudanças necessárias; (2) passar de um gerenciador de BD de um fornecedor para outro; (3) mudar de arquitetura de BD centralizada para cliente/servidor; (4) implementar interface de BD mais inteligente para o usuário e (5) integrar os BDs isolados.

Este trabalho propõe uma metodologia para a engenharia reversa de banco de dados relacional (BDR), a qual não exige as relações na terceira forma normal. Esta exigência limita a utilidade das metodologias existentes, uma vez que, na prática, muitas implementações violam as regras de normalização, devido à necessidade de otimização ou ao mau projeto. Os BDRs que desrespeitam as regras são os mais relevantes para a engenharia reversa.

Uma das contribuições importantes da metodologia proposta está na utilização da matriz Relações x Chaves Candidatas. Como a transposição de chaves representa as referências de uma relação a outra, a utilização da matriz simplifica a análise. A matriz torna o processo de coleta de informações mais racional e dirigido. Finalmente, a matriz permite estabelecer as regras para derivar o esquema conceitual de BDR, usando o modelo de entidades e relacionamentos.

(*) Bacharel, Mestre e Doutor em Administração de Empresas pela Faculdade de Economia, Administração e Contabilidade da Universidade de São Paulo. Professor do Departamento de Administração da FEA/USP. E-mail: takaoka@usp.br.

INTRODUÇÃO

Há várias motivações para fazer a engenharia reversa de BDs: (1) recuperar a descrição do conteúdo do BD que se perdeu ao longo do tempo devido a modificações para implementar mudanças necessárias; (2) passar de um gerenciador de BD de um fornecedor para outro; (3) mudar de arquitetura de BD centralizada para cliente/servidor; (4) implementar interfaces de BD mais inteligentes para o usuário e (5) integrar os BDs isolados. Para isso, é essencial entender o significado exato dos dados desses BDs existentes. Uma das formas de se obter este entendimento é descrevê-los através de um modelo conceitual de dados que permite representar melhor as semânticas inerentes à aplicação. Assim, será utilizado o modelo de entidades e relacionamentos (MER) ([5], [8], [10] e [11]).

Este trabalho trata da engenharia reversa de bancos de dados relacionais (BDRs), pois são mais populares. Além disso, vários estudos têm apontado a deficiência do modelo relacional (MR) na representação de semânticas da aplicação. Por exemplo, Schmid e Swenson [9] observa que a teoria do BDR não dá nenhuma indicação da maneira em que o domínio da aplicação pode ser representado por conjunto de relações. Kent [4] descreve as limitações do MR na representação das semânticas dos dados. As características do MR podem ser encontradas no Date [1].

A metodologia proposta é aplicável a BDRs que não estão na terceira forma normal (3FN). A exigência de as relações estarem na 3FN limita a utilidade das metodologias existentes, uma vez que, na prática, muitas implementações violam as regras de normalização devido a necessidade de otimização ou ao mau projeto. As implementações que desrespeitam as regras são as mais relevantes para a engenharia reversa.

Trabalhos Relacionados

Há vários trabalhos que tratam da engenharia reversa de BDRs. Por exemplo, Davis & Arora [2] descrevem uma metodologia para traduzir o esquema de BDR para o MER. Seu objetivo é gerar um esquema em ER invertível. Essa metodologia exige que as relações estejam na 3FN e que não haja homônimos e sinônimos. Navathe & Awong [6] propõem uma metodologia de conversão mais poderosa baseada no MER estendido. Sua abordagem não impõe a restrição de invertibilidade. Mas, requer também o pré-processamento para transformar as relações na 3FN e eliminar homônimos e sinônimos. A desvantagem da abordagem de pré-processamento é que pode exigir um volume muito grande de informações bem no início da conversão. Nem todas as informações podem estar disponíveis. Além disso, é difícil saber *a priori* as informações que devem ser coletadas, uma vez que a conversão requer muitas interações, tomando decisões sobre a interpretação dos dados à medida que o seu significado for compreendido. Já, Johannesson & Kalman [3] propõem uma metodologia que gera um esquema conceitual, o qual não está restrito às informações contidas no esquema de BDR, como nas metodologias anteriores. Como um esquema conceitual deve ter muito mais informações semânticas do que esquema de BDR, esta metodologia propõe formas de interações com os usuários para obter essas informações adicionais. Mas, apresenta a limitação de exigir que as relações estejam na 3FN. Finalmente, Premerlani & Blaha [7] propõem uma metodologia baseada no modelo orientado a objetos que é muito similar ao MER estendido. Uma das vantagens desta metodologia é que é aplicável a BDRs que não estão na 3FN. Outra contribuição importante desse trabalho está no relato das experiências obtidas através de vários estudos de casos de BDRs. Segundo este trabalho, a metodologia deveria permitir deduzir as estratégias que o projetista adotou na conversão do esquema conceitual. Muitas vezes, o projetista usa estratégia consistente, inclusive violações consistentes de regras de conversão. Assim, com uma abordagem puramente mecânica não será possível levar em consideração as transformações que o projetista empregou na implementação do esquema conceitual.

METODOLOGIA

Esta metodologia se baseia, principalmente, na análise da intenção do BDR e, quando necessário, na análise de sua extensão para investigar os dados propriamente ditos e na análise da aplicação. As informações coletadas são registradas na matriz Relações x Chaves Candidatas. As relações são listadas nas colunas e as chaves candidatas, nas linhas. Nas interseções de linhas com colunas, denominadas células, são registradas a função da chave (primária, alternativa ou externa), a sua composição e a sua origem (nativa ou transposta). A razão de não registrar explicitamente as chaves externas é que são normalmente difíceis de serem identificadas. Além disso, sua função é fazer referência à relação da qual foi transposta. Desta forma, será registrada apenas a origem e o destino da transposição. Como a transposição de chaves representa as referências de uma relação a outra, a utilização da matriz simplifica não só o registro dessas referências, mas principalmente a sua análise, já que facilita a sua visualização.

Visando facilitar a explicação, serão utilizadas as seguintes notações:

- C_j vetor associado à relação da coluna j;
- L_i vetor associado à chave da linha i;
- x_{ij} elemento da interseção da linha i com a coluna j;
- r_j relação da coluna j;
- c_i chave da linha i;
- c_k^i chave c_k que pode ter sido transposta para formar a chave c_i;
- $v(c_i)$ conjunto de valores da chave c_i;
- $na(c_i)$ número de atributos da chave c_i.

A matriz admite os seguintes valores para o elemento x_{ij}:

- P indica que a chave c_i é primária da relação r_j;
- A indica que a chave c_i é alternativa da relação r_j;
- x indica que a chave c_i aparece como externa na relação r_j;
- t:k indica que a chave c_i foi transposta para a relação r_j como componente de sua chave candidata c_k;
- e:k indica que a chave c_i foi transposta como c_k para a relação de especialização r_j;
- s:k indica que a chave c_i foi transposta como c_k para a relação de subconjunto r_j.

Para a descrição da metodologia será usado o esquema de BDR da Figura 1. A matriz correspondente ao BDR do exemplo é a que consta da Matriz 1.

Os passos desta metodologia são: (1) Registrar as relações; (2) Registrar as chaves candidatas (primárias e alternativas); (3) Pesquisar as chaves transpostas; (4) Analisar a matriz; (5) Converter as relações em entidades e relacionamentos e (6) Aprimorar o esquema de entidades e relacionamentos. Observamos que estes passos devem ser executados de forma interativa. A seguir, serão detalhados cada um dos passos.

DEPARTAMENTO(Depto, Nome, Diretor, ...)
FUNCIONÁRIO(IdFunc, Nome, PIS, Depto, Categoria, Salário, TempoDep, ...)
ACIONISTA(CPF, Nome, Participação, ...)
FUNC_ACIONISTA(IdFunc, Acionista, ...)
ADMINISTRATIVO(IdFunc, ...)
TÉCNICO(IdFunc, Nível, ...)
PARTICIPAÇÃO(IdFunc, Projeto, Atividade, ...)
ATIVIDADE(CodAtiv, Descrição, ...)
PROJETO(IdProj, Nome, Descrição, Local, UF, Valor, Moeda, Cliente, Endereço, ...)
CLIENTE_VIP(IdCliente, Nome, Contato, Telefone, ...)
FINANCIAMENTO(IdProj, IdFinanc, NomeFinanc, Valor, ...)
COTAÇÃO(Pais, Cotação, ...)
ATIVIDADE_PROJETO(Proj, Ativ, NumProjAtiv, ...)
CRONOGRAMA(NumProjAtiv, DataI, DataT, ...)
FINANC_EXTERNO(IdFinanc, País, ...)
INVESTIMENTO(País, Ano, Montante, ...)

Obs.: As chaves primárias estão sublinhadas com a linha cheia e as chaves alternativas, com a linha pontilhada.

Figura 1

Passo 1. Registrar as Relações

Registre as relações nas colunas da matriz. Muitas vezes, uma relação é decomposta visando a otimização ou a dispersão de seus registros num ambiente distribuído. Antes de registrar as rela-

ções na matriz, agrupe numa única relação as relações particionadas horizontal ou verticalmente. As relações particionadas horizontalmente têm o mesmo esquema, isto é, a mesma intenção. Como as relações particionadas verticalmente têm a mesma chave primária, elas podem ser confundidas com as relações de uma hierarquia de generalização ou de subconjuntos. Pode ser necessário analisar o conjunto de valores de suas chaves para dirimir a dúvida.

Passo 2. Registrar as chaves candidatas

Registre as chaves candidatas das relações nas linhas da matriz. Em cada célula x_{ij}, indique o papel da chave candidata c_i na relação r_j. Isto é feito especificando **P** para as chaves primárias e **A** para as chaves alternativas. A razão de não se restringir apenas à identificação da chave primária, é que, muitas vezes, a chave alternativa é usada para fazer referência a outras relações. Portanto, a sua identificação é fundamental para descobrir as chaves transpostas.

As chaves candidatas podem ser facilmente identificadas procurando, por exemplo, as opções UNIQUE, PRIMARY KEY e UNIQUE INDEX no esquema. Observamos que a opção UNIQUE INDEX nem sempre está definida para todas as chaves alternativas. Os atributos que são chaves, muitas vezes, têm os seguintes prefixos: #, Id, Ident, Num, Cod, etc. Uma análise exploratória dos dados pode sugerir padrões que permitem identificá-las.

Passo 3. Pesquisar as chaves transpostas

Visto que a transposição de chaves representa as referências de uma relação a outra, é necessário identificar as chaves transpostas. A regra para garantir que essas referências existam e estejam compatíveis é conhecida como integridade referencial. Basicamente, há duas maneiras para implementar a integridade referencial. A implementação logicamente mais correta é aquela feita na definição de esquemas através da opção FOREIGN KEY do gerenciador. Outra forma é colocar a integridade referencial fora do gerenciador, construindo as restrições dentro de cada aplicação. Se o gerenciador no qual o banco de dados está implementado tiver a opção FOREIGN KEY no comando CREATE, basta analisar o esquema para identificar as chaves transpostas. Caso contrário, é necessário descobri-las, analisando o esquema (intenção), os dados (extensão) e as aplicações. Para identificá-las, tente resolver o problema de sinônimo e homônimo, comparando nomes, tipos de dados e/ou domínios de atributos. A análise, por exemplo, de junções de relações, definições de VIEWS e índices secundários usados pode, também, ser útil. O sentido da transposição nem sempre é óbvio, principalmente nas hierarquias de generalização e de subconjuntos. Considere as chaves c_i e c_k com o mesmo domínio. A chave c_i será a de destino da transposição, se $v(c_i) \subseteq v(c_k)$. Note-se que esta condição é necessária, mas não é suficiente. Pode ser necessário outras informações para entender seu significado exato. Em muitos casos, a análise do esquema é suficiente para identificar a origem e o destino das chaves. A análise dos dados deve ser feita apenas quando essa distinção não for óbvia. A seguir, serão listados os procedimentos para ajudar a identificação das chaves transpostas.

Análise de chaves primárias compostas. As chaves primárias de uma relação que são compostas por vários atributos são normalmente concatenação de chaves transpostas de outras relações. Suponha que a chave c_i da relação r_j é uma chave primária com mais de um atributo, isto é, $na(c_i)>1$. Procure na matriz as chaves c_k^i ($1 \le k \le n$ e $i \neq k$) a partir das quais a chave c_i pode ter sido formada. Registre **t:i** na célula x_{kj} para indicar que a chave c_k^i foi transposta para formar a chave c_i. Considere o seguinte exemplo:

ATIVIDADE_PROJETO(Proj, Ativ, NumProjAtiv, ...)
CRONOGRAMA(NumProjAtiv, DataI, DataT, ...)
ATIVIDADE(CodAtiv, Descrição, ...)
PROJETO(IdProj, Nome, Descrição, Local, UF, Valor, Moeda, Cliente, Endereço, ...)

Foi registrado o valor **t:16** nas células $x_{11,13}$ e $x_{12,13}$ para indicar que as chaves CodAtiv (c_{11}) e IdProj (c_{12}) foram transpostas para formar a

chave da relação ATIVIDADE_PROJETO (c_{16}) e o valor **t:18** na célula $x_{17,14}$ para indicar que a chave alternativa NumProjAtiv (c_{17}) foi transposta para formar a chave da relação CRONOGRAMA (c_{18}). Note-se que uma chave alternativa pode ser usada na transposição, desde que seja apenas um identificador alternativo e não uma chave transposta de outra relação. Neste exemplo, admitimos que $v(c_i) \subseteq v(c_k^i)$, isto é, ATIVIDADE-PROJETO.Proj ⊆ PROJETO.IdProj, ATIVIDADE_PROJETO.Ativ ⊆ ATIVIDADE.CodAtiv e CRONOGRAMA.NumProjAtiv ⊆ ATIVIDADE_PROJETO.NumProjAtiv. sem analisar os dados. Se esta conclusão não for óbvia, recomenda-se analisar os dados. Se $v(c_i) \supset v(c_k^i)$, a relação com a chave c_k^i pode ser um subconjunto da relação da qual a chave foi transposta para a relação r_j. Essa relação, a partir da qual o subconjunto foi derivado, pode não ter sido implementada por não ter atributos descritivos ou por terem sido incluídos em outra relação, desrespeitando a segunda forma normal (2FN). Neste caso, é necessário inclui-la na matriz e atualizar as referências. Considere o seguinte exemplo:

PROJETO(IdProj, Nome, Descrição, Local, UF, Valor, Moeda, Cliente, Endereço, ...)
FINANCIAMENTO(IdProj, IdFinanc, NomeFinanc, Valor, ...)
FINANC_EXTERNO(IdFinanc, País, ...)
FINANCIAMENTO.IdProj ⊆ PROJETO.IdProj
FINANC_EXTERNO.IdFinanc ⊆ FINANCIAMENTO.IdFinanc

Neste exemplo, a relação FINANCIAMENTO não está na 2FN. Assim, é necessário explicitar a relação FINANCIADOR (r_{17}) com a chave IdFinanc (c_{22}), normalizando a relação FINANCIAMENTO. Na célula $x_{22,11}$ foi registrado o valor **t:14** para indicar que a chave desta nova relação foi transposta para formar a chave da relação FINANCIAMENTO (c_{14}) e na célula $x_{22,15}$ foi registrado o valor **s:19** para indicar que a relação FINANC_EXTERNO, cuja chave é IdFinanc (c_{19}), é um subconjunto da relação FINANCIADOR. Note-se que na Matriz 1 a nova relação e sua chave estão entre parênteses para indicar que foram incluídas no esquema.

Análise de chaves primárias compostas inteiramente por chaves primárias de outras relações. A ocorrência de grupos de chaves primárias que se referenciam uma a outra pode ser indicativo da existência de hierarquia. Na hierarquia de generalização, há, normalmente, na relação de nível mais alto, um atributo cujos valores discriminam as relações de nível imediatamente abaixo. Esses valores, muitas vezes, são semelhantes ao nome das relações discriminadas. Suponha as chaves c_i e c_k^i , respectivamente das relações r_j e r_m. Se na relação r_m tiver o atributo cujo valor está discriminando a relação r_j, registre **e:i** no elemento x_{kj} para indicar que a relação r_m é uma generalização da relação r_j. Considere o seguinte exemplo:

FUNCIONÁRIO(IdFunc, Nome, PIS, Depto, Categoria, Salário, TempoDep, ...)
ADMINISTRATIVO(IdFunc, ...)
TÉCNICO(IdFunc, Nível, ...)
Categoria = {técnico, administrativo}

Como os valores do atributo Categoria da relação FUNCIONÁRIO discriminam as relações ADMINISTRATIVO e TÉCNICO, foram registrados nas células $x_{3,5}$ e $x_{3,6}$ os valores **e:8** e **e:9** respectivamente para indicar que a relação FUNCIONÁRIO é a generalização das relações ADMINISTRATIVO e TÉCNICO.

Na ausência do atributo discriminador, pode ser necessário analisar os dados para identificar os subconjuntos. Considere as chaves c_i e c_k^i , respectivamente das relações r_j e r_m. Se $v(c_i) \subseteq v(c_k^j)$, então a relação r_j é subconjunto da relação r_m. Neste caso, registre **s:i** na célula x_{kj} para indicar que a relação r_j é um subconjunto da relação r_m. Considere o seguinte exemplo:

FUNCIONÁRIO(IdFunc, Nome, PIS, Depto, Categoria, Salário, TempoDep, ...)
FUNC_ACIONISTA(IdFunc, CPF, ...)
FUNC_ACIONISTA.IdFunc ⊆ FUNCIONÁRIO.IdFunc

Neste exemplo, o valor **s:6** na célula $x_{3,4}$, está indicando que a relação FUNC_ACIONISTA, cuja chave é IdFunc (c_6), é um subconjunto da relação FUNCIONÁRIO.

Análise de chaves alternativas. A razão da complicação causada pelas chaves alternativas é que elas podem indicar um relacionamento 1:1 ou uma hierarquia de subconjuntos. No relacionamento binário 1:1, muitas vezes, a chave transposta pode aparecer como chave alternativa. Considere a relação r_j com a chave alternativa c_i e a relação r_m com a chave primária c_k^i. Se $v(c_i) \subseteq v(c_k^i)$, isto é, a chave c_k^i foi transposta para a relação r_j como chave alternativa c_i, indique com **t:i** a célula x_{kj}. Considere o seguinte exemplo:

DEPARTAMENTO(Depto, Nome, Diretor, ...)
FUNCIONÁRIO(IdFunc, Nome, PIS, Depto, Categoria, Salário, TempoDep, ...)
DEPARTAMENTO.Diretor ⊆ FUNCIONÁRIO.Idfunc

Neste exemplo, a célula $x_{3,1}$ foi preenchida com **t:2** para indicar que a chave alternativa Diretor (c_2) foi transposta da relação FUNCIONÁRIO (r_2).

Se a chave alternativa for um identificador natural alternativo da entidade representada pela relação e não de outro tipo de entidade, a relação pode ser um subconjunto da relação a partir da qual foi transposta a chave alternativa. Considere o seguinte exemplo:

ACIONISTA(CPF, Nome, Participação, ...)
FUNC_ACIONISTA(IdFunc, CPF, ...)
FUNC_ACIONISTA.CPF ⊆ ACIONISTA.CPF

Note-se que a chave alternativa CPF é um identificador natural alternativo do subconjunto de pessoas representado pela relação FUNC_ACIONISTA. Assim, neste exemplo, foi registrado o valor **s:7** na célula $x_{5,4}$ para indicar que a relação FUNC_ACIONISTA pode ser um subconjunto da relação ACIONISTA.

Se $v(c_i) \supset v(c_k^i)$, isto é, a chave alternativa c_i foi transposta para a relação r_m, então a relação r_m pode ser um subconjunto da relação r_j. Considere o exemplo abaixo:

FINANC_EXTERNO(IdFinanc, País, ...)
COTAÇÃO(País, Cotação, ...)
COTAÇÃO.País ⊆ FINANC_EXTERNO.País

Neste exemplo, foi registrado o valor **s:15** na célula $x_{20,12}$ para indicar que a relação COTAÇÃO pode ser um subconjunto da relação FINANC_EXTERNO.

Análise de chaves externas. Entende-se por chaves externas, as chaves que foram transpostas de outras relações, mas que não são candidatas. É conhecida também como chave estrangeira. A identificação de chaves externas pode ser difícil se os gerenciadores usados na implementação forem antigos, já que não têm a opção FOREIGN KEY. Normalmente, o relacionamento binário 1:N é implementado transportando a chave candidata da relação, cuja cardinalidade é 1, para a relação cuja cardinalidade é N. Sendo assim, qualquer atributo que não for descritivo é um candidato potencial. A análise de junções de relações, por exemplo, a definição de VIEWS e de índices secundários usados podem, também, ser úteis. Suponha a relação r_j com a chave externa *a* e a relação r_m com a chave c_k. Se a chave c_k foi transposta para a relação r_j como chave externa *a*, isto é, $v(a) \subseteq v(c_k)$, indique com **x** a célula x_{kj}. Considere o seguinte exemplo:

DEPARTAMENTO(Depto, Nome, Diretor, ...)
FUNCIONÁRIO(IdFunc, Nome, PIS, Depto, Categoria, Salário, TempoDep, ...)
FUNCIONÁRIO.Depto ⊆ DEPARTAMENTO.Depto

Neste exemplo, a célula $x_{1,2}$ foi preenchida com **x** para indicar que a chave Depto (c_1) foi transposta para a relação FUNCIONÁRIO, como uma chave externa. Muitas vezes, como o atributo do relacionamento pode ser que tenha sido transposto juntamente com a chave, é necessário verificar a dependência funcional dos atributos da relação do lado N. Por exemplo, o atributo TempoDep (tempo no departamento) é do relaciona-

mento entre DEPARTAMENTO e FUNCIONÁRIO e não do FUNCIONÁRIO.

Passo 4. Analisar a Matriz

Nesta etapa, pode ser necessário obter mais informações para dirimir ambigüidades. A seguir, serão listados alguns procedimentos para a verificação da matriz.

Verifique se há relações isoladas. Uma relação r_j com a chave primária c_i é isolada se tiver no vetor C_j apenas elementos com valor igual a P ou A, e no vetor L_i apenas elemento com valor igual a P. Em outras palavras, uma relação é isolada se não fizer referência a outras relações e não for referenciada por outras relações. A existência de relações isoladas nem sempre indica a falha na identificação das chaves externas, pois algumas relações podem ter sido implementadas especialmente para conter restrições e funções da aplicação. Como não fazem parte do esquema, é necessário exclui-las da matriz. Pode ser que as relações com as quais estão ligadas não foram incluídas no esquema por não terem atributos descritivos ou por terem seus atributos incluídos em outra relação, desrespeitando as regras de normalização. Nesses casos, é necessário incluí-las, explicitamente, na matriz e atualizar as referências. Considere, por exemplo, a relação CLIENTE_VIP. Pode-se notar que esta relação não faz referência a outras relações e não é referenciada por outras relações. O objeto CLIENTE a partir do qual a chave IdCliente (c_{13}) foi transposta não tem a relação correspondente por ter seus atributos incluídos na relação PROJETO, desrespeitando a 3FN. Assim, é necessário criar a relação CLIENTE, normalizando a relação PROJETO, e atualizar as referências. Neste exemplo, foi incluída a relação CLIENTE (r_{18}) com a chave Cliente (c_{23}). Na célula $x_{23,9}$ foi registrada **x** para indicar que a chave desta nova relação foi transposta para a relação PROJETO e na célula $x_{23,10}$ foi registrada **s:13** para indicar que a relação CLIENTE_VIP é um subconjunto da relação CLIENTE.

*Verifique se a chave primária composta de várias chaves transpostas está completa***.** Para tanto, basta verificar se o número de atributos da chave primária é igual à soma dos números de atributos das chaves transpostas. Considere a matriz a seguir:

	...	r_j	r_{j+1}	r_{j+2}	r_{j+3}	r_{j+4}	r_{j+5}	...
:								
c_i		P						
c_{i+1}		t:i	P					
c_{i+2}								
c_{i+3}		t:i			P			
c_{i+4}								
c_{i+5}		t:i					P	
:								

Observe-se que a chave c_i da relação r_j é formada pelas chaves c_{i+1}, c_{i+3} e c_{i+5} transpostas das relações r_{j+1}, r_{j+3} e r_{j+5} respectivamente. Verifique se o valor de na(c_i) é igual à soma dos valores de na(c_{i+1}), na(c_{i+3}) e na(c_{i+5}). Se a soma for menor, há duas possibilidades:

1. Os atributos são de "relacionamento". Neste caso, é necessário transformá-los em chaves transpostas, criando relações tendo estes atributos como chaves primárias. Vimos que as entidades sem os atributos descritivos nem sempre são implementadas como uma relação. Neste caso, deve-se acrescentar a relação na matriz e atualizar as referências. Um exemplo que ilustra este caso é a relação CRONOGRAMA. Como as chaves DataI e DataT são de "relacionamento", é necessário criar a relação DATA com a chave Data e registrar o conjunto {**t:18**, **t:18**} na célula

$x_{25,14}$ para indicar que esta chave foi transposta para a relação CRONOGRAMA.

2. Os atributos apenas complementam a chave para garantir a unicidade do registro. Em outras palavras, a relação depende da relação a partir da qual a chave foi transposta (dependência de chave ou de identificação). A relação INVESTIMENTO é um exemplo deste caso. O atributo Ano juntamente com a chave País transposta da relação FINANC_EXTERNO formam a chave primária da relação INVESTIMENTO.

Se a soma for maior, pode ser devido a existência de mais de uma origem da chave transposta ou devido ao problema de fecho transitivo originado pela combinação de hierarquia com o relacionamento. Estes casos serão discutidos nos itens específicos.

Verifique se deixou de explicitar uma entidade. Vimos que uma relação pode não ter sido incluída no esquema por não ter atributos descritivos ou por ter seus atributos incluídos em outra relação, desrespeitando as regras de normalização. Considere a matriz abaixo:

	...	r_j	r_{j+1}	r_{j+2}	r_{j+3}	r_{j+4}	r_{j+5}	...
:								
c_i		P						
c_{i+1}		A		s:i+3	t:i+4			
c_{i+2}								
c_{i+3}				P				
c_{i+4}					P			
:								

Note-se que a chave alternativa c_{i+1} foi transposta para as relações r_{j+2} e r_{j+3}. Se esta chave c_{i+1} for uma chave que foi transposta de outra entidade e não um identificador natural alternativo da relação r_j, é necessário explicitar essa entidade através de uma nova relação na matriz e atualizar as referências. Considere as relações FINANC_EXTERNO e COTAÇÃO. A chave alternativa País (c_{20}) da relação FINANC_EXTERNO que foi transposta para a relação COTAÇÃO não é um identificador natural alternativo, mas um identificador da entidade PAÍS. Assim, é necessário criar a relação PAÍS e atualizar as referências. Neste exemplo, foi incluída a relação PAÍS (r_{19}) com a chave País (c_{24}). Na célula $x_{24,12}$ foi registrada **s:15** para indicar que a relação COTAÇÃO é um subconjunto da relação PAÍS e o conteúdo da célula $x_{20,12}$ foi excluído, uma vez que a relação COTAÇÃO não é um subconjunto da relação FINANC_EXTERNO.

Verifique se a chave transposta tem mais de uma origem. A possibilidade de uma chave transposta ter mais de uma origem deve-se ao fato de um esquema conceitual poder representar mais informações semânticas do que um esquema relacional. Considere os esquemas da Figura 2. Note-se que os dois esquemas conceituais resultam no mesmo esquema relacional. Assim, na conversão do esquema relacional para o conceitual é necessário informações adicionais para dirimir a dúvida. As chaves da relação S têm mais de uma origem. No primeiro esquema, as chaves das relações C e R foram transpostas para a relação S. Já no segundo, as chaves das relações A, B e C foram transpostas para a relação S. Para ilustrar, vamos analisar a relação PARTICIPAÇÃO (r_7). A soma dos atributos das chaves indicadas como origem de transposição é maior do que o número de atributos da chave de destino. Esta diferença a maior deve-se ao fato das chaves Proj e Ativ poderem ter sido transpostas da relação ATIVIDADE_PROJETO ou das relações PROJETO e ATIVIDADE. Para dirimir a ambigüidade é necessário mais informações semânticas. Vamos supor que as informações adicionais revelaram que as chaves foram transpostas da relação ATIVIDADE_PROJETO. Neste caso, é necessário atualizar a matriz, excluindo os valores registrados nas células $x_{11,7}$ e $x_{12,7}$.

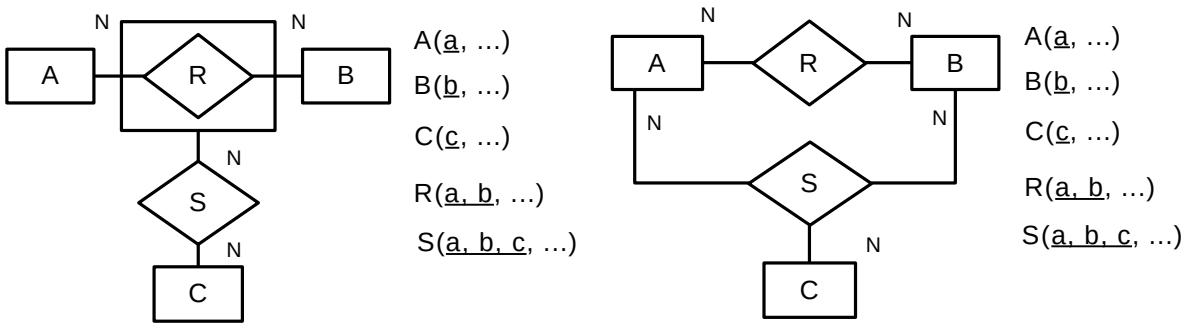


Figura 2

Verifique se há o problema de fecho transitivo. A combinação de hierarquia de generalização e de subconjuntos com o relacionamento pode dar origem a fecho transitivo. Considere a matriz abaixo:

	...	r_j	r_{j+1}	r_{j+2}	r_{j+3}	r_{j+4}	...
:							
c_i		P	e:i+1	e:i+2	x ou t:i+3		
c_{i+1}			P		x ou t:i+3		
c_{i+2}				P	x ou t:i+3		
c_{i+3}					P		
:							

Observe-se que há referências redundantes, pois a relação r_{j+3} faz referência a todas as relações (r_j, r_{j+1} e r_{j+2}) que formam uma hierarquia de generalização. Apenas com as informações do esquema, não é possível determinar de qual das relações da hierarquia foi transposta a chave estrangeira. Neste caso, é necessário obter informações semânticas adicionais. Vamos supor que a relação r_{j+3} faz referência apenas à relação r_{j+1} e nunca à relação $r_{j+2.}$ Neste caso, o conteúdo dos elementos $x_{i,j+3}$ e $x_{i+2,\ j+3}$ devem ser excluídos. No exemplo, a combinação da hierarquia formada pelas relações FUNCIONÁRIO, ADMINISTRATIVO, TÉCNICO e FUNC_ACIONISTA com o relacionamento PARTICIPAÇÃO está dando origem a fecho transitivo. Vamos supor que apenas os técnicos participam do relacionamento. Neste caso, é necessário excluir o conteúdo das células $x_{3,7}$, $x_{6,7}$ e $x_{8,7}$.

Verifique se há ligações redundantes na hierarquia de subconjuntos. O procedimento para identificar a hierarquia de subconjuntos pode ter introduzido ligações redundantes na hierarquia de subconjuntos. Considere a matriz que se segue:

	...	r_j	r_{j+1}	r_{j+2}	r_{j+3}	r_{j+4}	...
:							
c_i		P	s:i+1	s:i+2	s:i+3	s:i+4	
c_{i+1}			P		s:i+3	s:i+4	
c_{i+2}				P			
c_{i+3}					P		
c_{i+4}						P	
:							

A análise isolada do vetor C_{j+3} revela que a relação r_{j+3} é um subconjunto das relações r_j e r_{j+1}. Mas, o vetor C_{j+1} indica que a relação r_{j+1} é também um subconjunto da relação r_j. Se a relação r_{j+1} é um subconjunto da relação r_j e a relação r_{j+3} é um subconjunto da relação r_{j+1}, então a relação r_{j+3} é um subconjunto da relação r_j por transitividade. O mesmo problema ocorre com as relações r_j, r_{j+1} e r_{j+4}. Portanto, as indicações de que as relações r_{j+3}e r_{j+4} são subconjuntos da relação r_j devem ser excluídas.

Verifique se há transposição dupla da chave candidata. Há casos em que um relacionamento 1:1 é implementado transpondo a chave primária de uma para outra e vice-versa. Considere a matriz abaixo:

	...	r_j	r_{j+1}	r_{j+2}	r_{j+3}	r_{j+4}	...
:							
c_i		P	t:i+3				
c_{i+1}		A					
c_{i+2}		t:i+2	P				
c_{i+3}			A				
:							

Note-se que a relação r_j tem como chave alternativa a chave c_{i+1} transposta da relação r_{j+1} e a relação r_{j+1} tem como chave alternativa a chave c_i transposta da relação r_j. Como cada chave alternativa corresponde a um relacionamento, um dos relacionamentos é redundante. Portanto, deve ser excluído. Note-se que esta redundância é uma indicação de que o relacionamento é 1:1.

Verifique se as chaves externas de uma relação transpostas de uma mesma relação desempenham papeis distintos. Uma relação que tem várias chaves externas transpostas de uma mesma relação deve ser analisada. Pode ser que uma entidade, ligada por apenas um relacionamento N:N com outra entidade tenha sido modelada por vários relacionamentos N:1 com mesmo papel. É uma forma de otimização empregada na implementação do BD motivada pelo desejo de melhorar o desempenho. Considere a seguinte matriz:

	...	r_j	r_{j+1}	r_{j+2}	r_{j+3}	r_{j+4}	...
:							
c_i		P					
c_{i+1}							
c_{i+2}		x, x, x, x, x		P			
:							

Note-se que a relação r_j tem várias chaves externas transpostas da relação r_{j+2}. Se for constatado que se trata de um artifício empregado visando a otimização, transforme num único relacionamento N:N.

Passo 5. Converter as relações em entidades e relacionamentos

Após verificar a completeza da matriz, analisar os vetores da coluna para converter as relações em entidades e relacionamentos. As regras de conversão estão baseadas no conteúdo do vetor. Para que uma regra de conversão possa ser aplicada, basta que o vetor em análise tenha o conteúdo estabelecido por esta regra. Assim, mais de uma conversão pode ser aplicada a um vetor dependendo do seu conteúdo. A seguir, são listadas as regras de conversão:

C1. $\mathbf{C_j=[..., x_{ij}=P, ...]}$

Uma relação com a chave primária formada pelos atributos que não são chaves transpostas, isto é, não há no vetor C_j elementos com valor **t:i** ou **e:i** ou **s:i**, pode ser convertida numa entidade. Note-se que esta regra pode ser aplicada aos vetores correspondentes às relações ACIONISTA, ATIVIDADE, FINANCIADOR, CLIENTE, PAÍS e DATA.

C2. $\mathbf{C_j=[..., x_{ij}=P, ..., x_{kj}=t:i, ..., x_{lj}=t:i, ...], onde\ na(c_i)=na(c_k)+ ... +na(c_l)}$

Uma relação cuja chave primária é a concatenação de duas ou mais chaves transpostas pode ser convertida num relacionamento que associa as entidades referenciadas. É importante observar que as entidades não precisam ser necessariamente distintas (auto-relacionamento). Todo elemento multivalorado, indica a existência de auto-relacionamento. Por exemplo, suponha um vetor C_j=[..., x_{ij}=**P**, ..., x_{kj}=**t:i**, ..., x_{lj}={**t:i**, **t:i**}, ...], onde $na(c_i)=na(c_k)+ ... +na(c_l)+na(c_l)$. Como o elemento x_{lj} é multivalorado, existe um auto-relacionamento ligando os elementos da entidade correspondente à relação cuja chave é c_l. No exemplo, os vetores correspondentes às relações PARTICIPAÇÃO, FINANCIAMENTO, ATIVIDADE_PROJETO e CRONOGRAMA satisfazem a esta regra. Além disso, se a chave c_k que foi transposta para formar a chave c_i for de relacionamento, este relacionamento deve ser reinterpretado como uma entidade. Vamos chamar esta conversão de ***C2´***. Note-se que no exemplo o relacionamento ATIVIDADE_PROJETO foi abstraído como uma entidade agregada. No diagrama, esta abstração é representada envolvendo o losango com um retângulo.

C3. $\mathbf{C_j=[..., x_{ij}=P, ..., x_{kj}=t:i, ..., x_{lj}=t:i, ...], onde\ na(c_i)\geq na(c_k)+ ... +na(c_l)}$

Uma relação cuja chave primária é composta pelas chaves transpostas e atributos pode ser convertida numa entidade com dependência de chave com a entidade da qual a chave foi transposta. Esta última entidade pode ser um relacionamento entre entidades denominada entidade agregada. Observe-se que esta regra pode ser aplicada ao vetor correspondente à relação INVESTIMENTO.

C4. $\mathbf{C_j=[..., x_{ij}=P, ..., x_{kj}=x, ...]}$

Uma relação com chave externa, isto é, chave transposta que não faz parte da chave candidata pode ser convertida numa entidade e relacionamento N:1, com a entidade referenciada pela chave externa, isto é, com a relação cuja chave é c_k. No exemplo, esta regra pode ser aplicada aos vetores correspondentes à relação FUNCIONÁRIO e PROJETO.

C5. $\mathbf{C_j=[..., x_{ij}=P, ..., x_{kj}=A, ..., x_{lj}=t:k, ...]}$

Uma relação com a chave alternativa c_k transposta de outra relação pode ser convertida numa entidade e um relacionamento 1:1 com a entidade correspondente à relação cuja chave é c_l. No exemplo, esta regra pode ser aplicada aos vetores correspondentes às relações DEPARTAMENTO e FINANC_EXTERNO.

C6. $\mathbf{C_j=[..., x_{ij}=P, ..., x_{kj}=e:i, ...]}$

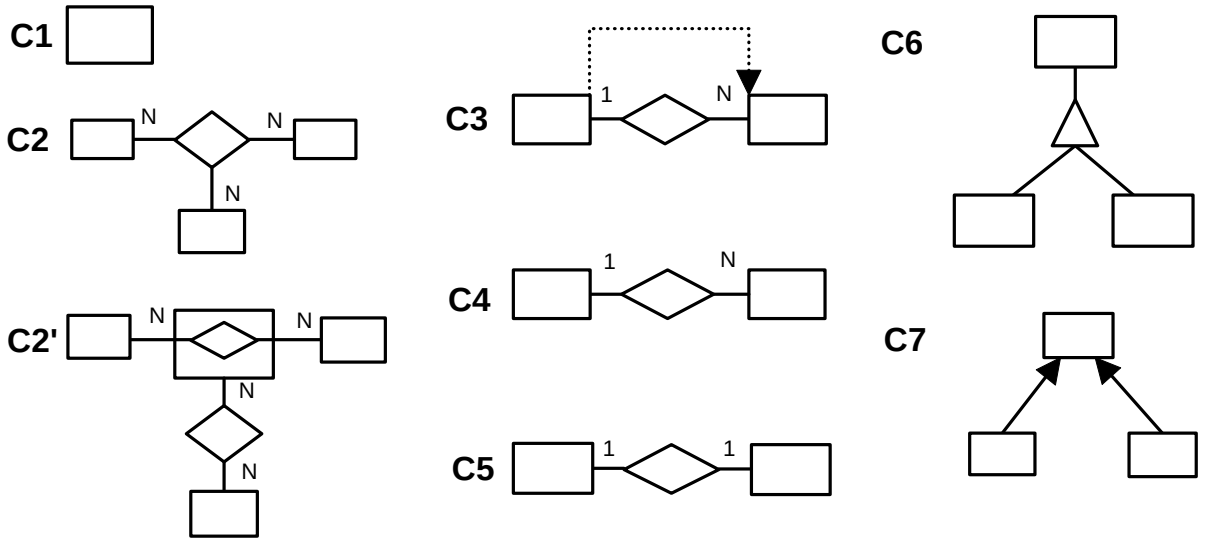
Esta relação deve ser convertida numa entidade especializada da entidade correspondente à relação cuja chave é c_k. No exemplo, esta regra pode ser aplicada aos vetores correspondentes às relações ADMINISTRATIVO e TÉCNICO.

C7. **C_j=[..., x_{ij}=P ou A, ..., x_{kj}=s:i, ...]**

Esta relação deve ser convertida num subconjunto da entidade correspondente à relação cuja chave é c_k. No exemplo, esta regra pode ser aplicada aos vetores correspondentes às relações FUNC_ACIONISTA, CLIENTE_VIP, COTAÇÃO e FINANC_EXTERNO.

A Figura 3 resume os diagramas de entidades e relacionamentos correspondentes às conversões propostas.

Figura 3



Como o modelo relacional não explicita o significado dos relacionamentos criados nas conversões ***C3***, ***C4*** e ***C5***, pode ser necessário informações adicionais para definir seu significado exato. A Figura 4 mostra o esquema conceitual correspondente ao esquema relacional do exemplo.

Relações

20	(DATA)
19	(PAÍS)
18	(CLIENTE)
17	(FINANCIADOR)
16	INVESTIMENTO
15	FINANC_EXTERNO
14	CRONOGRAMA
13	ATIVIDADE_PROJETO
12	COTAÇÃO
11	FINANCIAMENTO
10	CLIENTE_VIP
9	PROJETO
8	ATIVIDADE
7	PARTICIPAÇÃO
6	TÉCNICO
5	ADMINISTRATIVO
4	FUNC_ACIONISTA
3	ACIONISTA
2	FUNCIONÁRIO
1	DEPARTAMENTO

	Chaves Candidatas	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
1	Depto	P	x																		
2	Diretor	A																			
3	IdFunc	t:2	P		s:6	e:8	e.9	t:10													
4	PIS		A																		
5	CPF			P	s:7																
6	IdFunc				P			t:10													
7	Acionista				A																
8	IdFunc					P		t:10													
9	IdFunc						P	t:10													
10	IdFunc, Proj, Ativ							P													
11	CodAtiv							t:10	P					t:16							
12	IdProj							t:10		P		t:14		t:16							
13	IdCliente										P										
14	IdProj, IdFinanc											P									
15	Pais												P								
16	Proj, Ativ							t:10						P							
17	NumProjAtiv													A	t:18						
18	NumProjAtiv, DataI, DataT														P						
19	IdFinanc															P					
20	País												s:15			A					
21	Pais, Ano																P				
22	(IdFinanc)											t:14				s:19		P			
23	(Cliente)									x	s:13								P		
24	(País)												s:15			t:20	t:21			P	
25	(Data)														t:18 t:18						P
	Conversão	C5	C4	C1	C7	C6	C6	C2	C1	C4	C7	C2	C7	C2´	C2	C5	C3	C1	C1	C1	C1
																C7					

Observações: ☐ *Incluídos* ▒ *Excluídos*

Matriz 1

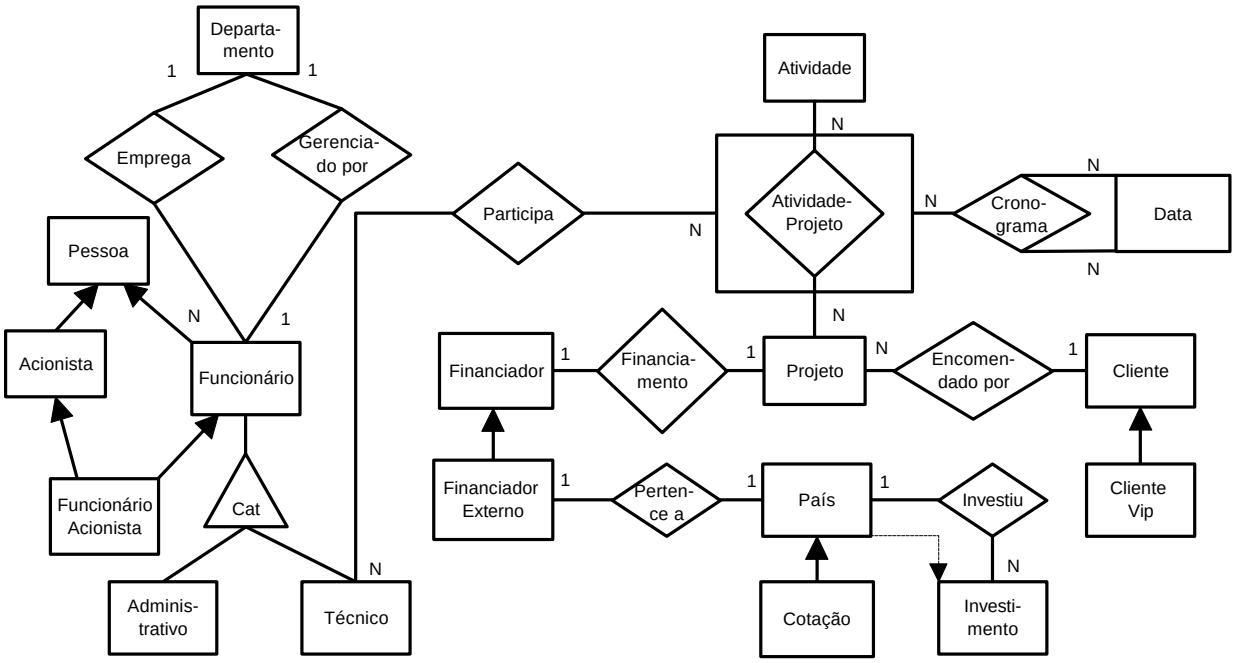


Figura 4

Passo 6. Aprimorar o esquema de entidades e relacionamentos

Como o objetivo da engenharia reversa é produzir o esquema conceitual e não simplesmente mudar a sintaxe do esquema, é necessário aprimorar o esquema obtido com base nas informações semânticas e regras de boa representação. Alguns exemplos de representação inadequada são:

1. Relacionamento N:N implementado como N:1 através de vetores de tamanho fixo. Esta solução pode ser eficiente no processamento, mas é inadequado em termos de clareza.
2. Hierarquia de generalização escondida. Muitas vezes, a classe genérica não é implementada. Os atributos comuns são embutidos nas classes especializadas. Outras vezes, as relações especializadas não são implementadas. Todos os atributos (comuns e especializados) são representados numa só relação. A análise de dados pode revelar estes casos.
3. Relacionamento entre mesmo objeto. Por exemplo, uma pessoa (objeto) que desempenha o papel de acionista e funcionário ao mesmo tempo é representada como duas entidades (respectivamente ACIONISTA e FUNCIONÁRIO) relacionadas através do relacionamento É_TAMBÉM. A representação correta é modelar ACIONISTA e FUNCIONÁRIO como sendo subconjuntos de PESSOA, e criar um subconjunto FUNCINÁRIO_ACIONISTA como sendo subconjunto de ACIONISTA e FUNCIONÁRIO (vide Figura 4).

CONCLUSÃO

Neste trabalho, foi proposta uma metodologia para a engenharia reversa de BDRs que não exige as relações na 3FN. O modelo escolhido foi o MER estendido que permite modelar a hierarquia de generalização/subconjuntos, entidades com dependência de chave e entidade agregada.

Uma das contribuições importantes desta metodologia é a utilização da matriz Relações x Chaves Candidatas. Em primeiro lugar, como a transposição de chaves candidatas e transpostas representa as referências de uma relação a outra,

a utilização da matriz simplifica a sua análise, já que facilita a visualização dessas referências. Em segundo lugar, através da matriz é possível analisar o esquema como um todo e identificar as informações que devem ser coletadas e analisadas à medida que forem necessárias, tornando o processo de coleta de informações mais racional e dirigido. Outra vantagem de se ter a visão do todo é que pode ser possível deduzir as estratégias que o projetista adotou na implementação do BDR. Finalmente, a matriz permite estabelecer regras para derivar o esquema conceitual do BDR.

Neste trabalho, não foi discutido se a metodologia proposta produz sempre um esquema conceitual equivalente, no sentido de existir uma transformação inversa, que ao ser aplicada ao esquema conceitual gera as relações originais, uma vez que o objetivo da engenharia reversa é descrever a intenção do BDR e não mudar simplesmente a sintaxe do esquema.

REFERÊNCIA BIBLIOGRÁFICA

DATE, C. J. *Introdução a Sistemas de Banco de Dados*. Editora Campus, Rio de Janeiro, (1984). (1)

DAVIS, K. H. & **ARORA**, A. K. *Converting a Relational Database Model into a Entity-Relationship Model.* Seventh International Conference on Entity-Relatioship Approach, (1987). (2)

JOHANNESSON, Paul & **KALMAN**, Katalin. *A Method for Translating Relational Schemas into Conceptual Schemas.* University of Stockholm, SYSLAB Report No 69, (July 1989). (3)

KENT, William. *DATA and REALITY*. North-Holland Publishing Company, (1978). (4)

KORTH, Henry F. & **SILBERSCHATZ**, Abraham. *Sistemas de Bancos de Dados.* Editora McGraw-Hill, Ltda., (1989). (5)

NAVATHE, S. B. & **AWONG**, A. M. *Abstracting relational and hierarquical data with a semantic data model*. Proceedings of the Sixth International Conference on Entity-Relationship Approach, (1987). (6)

PREMERLANI, William J. & **BLAHA**, Michael R. *An Approach for Reverse Engineering of Relational Databases.* Communications of the ACM, 37, 5, (May 1994), pp. 42-49. (7)

SCHEER, A.-W. *Business Process Engineering - Reference Models for Industrial Enterprises.* Springer-Verlag, Berlim Heidelberg, (1994). (8)

SCHMID, H. A. & **SWENSON**, J. R. *On the semantics of the relational data model.* Proceedings of the 1975 SIGMOD Conference, (1975). (9)

SETZER, Valdemar W. *Projeto Lógico e Projeto Físico de Banco de Dados.* V Escola de Computação, Belo Horizonte, (1986). (10)

TEOREY, T. J. et alli. *A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model.* ACM Computing Surveys, 2, (1986). (11)